Bandwidth Allocation Method and Apparatus for Fixed Wireless Networks

5

BACKGROUND OF THE INVENTION

TECHNICAL FIELD

10   The invention relates to bandwidth allocation in wireless networks. More particularly, the invention relates to optimizing bandwidth allocation for a fixed wireless network using a simple centralized algorithm to create a fair and efficient, high-performance bandwidth allocation schedule.

15   ### DESCRIPTION OF THE PRIOR ART

The allocation of bandwidth on demand in wireless access networks is important to provide high call-carrying capacity yet prevent degradation in the quality of a link due to interference arising from a number of simultaneous transmissions. The problem of

20   bandwidth allocation is one of allocation of radio resources for the duration of a transmission in both the uplink and downlink directions. The capacity of any cell is reached when the current resource utilization prevents the admission of additional calls, even at the lowest rate.

25   Wireless networks are expected to support real-time interactive multimedia traffic and must be able, therefore, to provide their users with Quality-of-Service (QoS) guarantees. Although the QoS provisioning problem arises in wireline networks as well, mobility of hosts and scarcity of bandwidth makes QoS provisioning a

1

challenging task in wireless networks. One problem with optimizing bandwidth allocation in a fixed wireless network concerns that fact that, given the constraints of the interference matrix and a list of bandwidth requests, it is difficult to find a schedule that makes optimal use of total network capacity and fairly satisfies

5   bandwidth requests.

An equivalent problem is that of finding an optimal schedule that satisfies all requests using the least amount of network resources where, if the average activity concurrency is defined as the average number of concurrent active links of a

10   schedule, then the optimal schedule is the one having the highest average activity concurrency.

It would be advantageous to provide a method and apparatus for optimizing bandwidth allocation for a fixed wireless network using a simple centralized algorithm

15   to create a fair and efficient, high-performance bandwidth allocation schedule.

SUMMARY OF THE INVENTION

The invention provides a method and apparatus for optimizing bandwidth allocation

20   for a fixed wireless network using a simple centralized algorithm to create a fair and efficient, high-performance bandwidth allocation schedule. A simple, fair, good-performance bandwidth allocation algorithm for wireless networks is presented. Using a matrix of interlink interference and a list of links' bandwidth requests, the algorithm can schedule link activities to obtain non-collision transmissions. All

25   bandwidth requests are served fairly and near-optimally based on the heuristic algorithm. Bandwidth granted for each link is prorated based on its requested bandwidth, total requested bandwidth in the network, and network capacity. The algorithm can be used for centralized bandwidth allocation and works with any network topology, including mesh networks.

30

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 is a tree diagram that shows a network having eleven nodes and twenty directional links;

Figure 2 is a block schematic diagram showing a system for bandwidth allocation in a fixed wireless network according to the invention; and

Figure 3 is a flow diagram showing an algorithm for bandwidth allocation in a fixed wireless network according to the invention.

DETAILED DESCRIPTION OF THE INVENTION

The invention provides a method and apparatus for optimizing bandwidth allocation for a fixed wireless network using a simple centralized algorithm to create a fair and efficient, high-performance bandwidth allocation schedule. Bandwidth allocation is typically based upon the requests of each individual link, all of the bandwidth requests in the network, link capacity, and inter-link interference. Thus, knowledge of the whole network, in terms of interference, eligible links, and link bandwidth requests is needed to create a bandwidth allocation schedule.

In the presently preferred embodiment of the invention, only one entity, referred to herein as a hub, needs this global view to compute the bandwidth allocation schedule. The bandwidth allocation schedule does not contain the actual slot assignment for each node. Rather, each node computes a slot assignment using a deterministic algorithm. Each grant in the bandwidth allocation schedule is a small integer. The bandwidth allocation schedule is not required to be sent periodically. The bandwidth allocation schedule is sent only when it changes. Hence, only a small amount of control traffic is used for dispatching the bandwidth allocation schedule.

Network model

For purposes of the discussion herein, a network is defined as a set of links between nodes. For example, the uni-directional link between Node I and Node J is called l_{ij}.

Suppose that there are N nodes and M directional links (l_{ij} and l_{ji} are considered different links) in a network. The interference between links in the network determines which links in the network can operate simultaneously. In other words, if a link l_{ij} is active there exists a set of links L_{ij} which cannot all be active at the same time. The set of all links L_{ij} in the network constitute the interference matrix of the network.

For purposes of the discussion herein, the degree of interference $\alpha(l_{ij}, L)$ of a directional link l_{ij} in a set L of links is defined as the number of links in set L that cannot be active due to interference while link l_{ij} is active.

The bandwidth needed by links to carry actual traffic over a specific time period is represented herein as a set of link bandwidth requests. The request may be zero. In that case, no traffic is to be carried over the link. Because link capacities vary depending on various link parameters, bandwidth requests are expressed in unit of credits, not bps (bits/sec). A credit is a unit the resource bandwidth allocation algorithm uses to maintain fair bandwidth distribution between links. It is the result of normalization of requested bandwidth, in terms of bps, with respect to the corresponding link capacity.

Example 1

4

Figure 1 is a tree diagram that shows a network having eleven nodes and twenty directional links:

5     $\{\ l_{0,1},\ l_{1,0},\ l_{0,2},\ l_{2,0},\ l_{1,3},\ l_{3,1},\ l_{1,4},\ l_{4,1},\ l_{2,5},\ l_{5,2},\ l_{2,6},\ l_{6,2},\ l_{2,7},\ l_{7,2},\ l_{4,8},\ l_{8,4},\ l_{5,9},\ l_{95},\ l_{6,10},\ l_{10,6}\}$

Suppose that the set of links $L_{0,1}$ that gets interference, *i.e.* that cannot be active while link $l_{0,1}$ is active, is:

10     $L_{0,1} = \{\ l_{1,0},\ l_{0,2},\ l_{2,0},\ l_{1,3},\ l_{3,1},\ l_{1,4},\ l_{4,1},\ l_{5,9},\ l_{8,4}\}$

Similarly, suppose there are the following interference sets:

$L_{1,0} = \{\ l_{0,1},\ l_{0,2},\ l_{2,0},\ l_{1,3},\ l_{3,1},\ l_{1,4},\ l_{4,1},\ l_{9,5},\ l_{4,8}\}$

15     $L_{0,2} = \{\ l_{1,0},\ l_{0,1},\ l_{2,0},\ l_{2,5},\ l_{5,2},\ l_{2,6},\ l_{6,2},\ l_{2,7},\ l_{7,2},\ l_{6,10}\}$

$L_{2,0} = \{\ l_{1,0},\ l_{0,1},\ l_{0,2},\ l_{2,5},\ l_{5,2},\ l_{2,6},\ l_{6,2},\ l_{2,7},\ l_{7,2},\ l_{10,6}\}$

$L_{1,3} = \{\ l_{3,1},\ l_{1,4},\ l_{4,1},\ l_{0,1},\ l_{1,0}\}$

$L_{3,1} = \{\ l_{1,3},\ l_{1,4},\ l_{4,1},\ l_{0,1},\ l_{1,0}\}$

$L_{1,4} = \{\ l_{4,1},\ l_{1,3},\ l_{3,1},\ l_{1,0},\ l_{0,1},\ l_{4,8},\ l_{8,4},\ l_{2,5},\ l_{7,2}\}$

20     $L_{4,1} = \{\ l_{1,4},\ l_{1,3},\ l_{3,1},\ l_{1,0},\ l_{0,1},\ l_{4,8},\ l_{8,4},\ l_{5,2},\ l_{2,7}\}$

$L_{2,5} = \{\ l_{5,2},\ l_{0,2},\ l_{2,0},\ l_{2,6},\ l_{6,2},\ l_{2,7},\ l_{7,2},\ l_{5,9},\ l_{9,5},\ l_{1,4}\}$

$L_{5,2} = \{\ l_{2,5},\ l_{0,2},\ l_{2,0},\ l_{2,6},\ l_{6,2},\ l_{2,7},\ l_{7,2},\ l_{5,9},\ l_{9,5},\ l_{4,1}\}$

$L_{2,6} = \{\ l_{6,2},\ l_{0,2},\ l_{2,0},\ l_{2,5},\ l_{5,2},\ l_{2,7},\ l_{7,2},\ l_{6,10},\ l_{10,6}\}$

$L_{6,2} = \{\ l_{2,6},\ l_{0,2},\ l_{2,0},\ l_{2,5},\ l_{5,2},\ l_{2,7},\ l_{7,2},\ l_{6,10},\ l_{10,6}\}$

25     $L_{2,7} = \{\ l_{7,2},\ l_{0,2},\ l_{2,0},\ l_{2,5},\ l_{5,2},\ l_{2,6},\ l_{6,2},\ l_{4,1}\}$

$L_{7,2} = \{\ l_{2,7},\ l_{0,2},\ l_{2,0},\ l_{2,5},\ l_{5,2},\ l_{2,6},\ l_{6,2},\ l_{1,4}\}$

$$L_{4,8} = \{\ I_{8,4}, I_{1,4}, I_{4,1}, I_{1,0}\ \}$$

$$L_{8,4} = \{\ I_{4,8}, I_{1,4}, I_{4,1}, I_{0,1}\ \}$$

$$L_{5,9} = \{\ I_{9,5}, I_{2,5}, I_{5,2}, I_{0,1}\ \}$$

$$L_{9,5} = \{\ I_{5,9}, I_{2,5}, I_{5,2}, I_{1,0}\ \}$$

5 $\qquad L_{6,10} = \{\ I_{10,6}, I_{2,6}, I_{6,2}, I_{0,2}\ \}$

$$L_{10,6} = \{\ I_{6,10}, I_{2,6}, I_{6,2}, I_{2,0}\ \}$$

Equivalently, the interference can be expressed using the interference matrix I shown in Table 1 below.

10

Table 1. Interference Matrix I

	$I_{0,1}$	$I_{1,0}$	$I_{0,2}$	$I_{2,0}$	$I_{1,3}$	$I_{3,1}$	$I_{1,4}$	$I_{4,1}$	$I_{2,5}$	$I_{5,2}$	$I_{2,6}$	$I_{6,2}$	$I_{2,7}$	$I_{7,2}$	$I_{4,8}$	$I_{8,4}$	$I_{5,9}$	$I_{9,5}$	$I_{6,10}$	$I_{10,6}$
$I_{0,1}$		1	1	1	1	1	1	1								1	1			
$I_{1,0}$	1		1	1	1	1	1	1							1			1		
$I_{0,2}$	1	1		1					1	1	1	1	1	1					1	
$I_{2,0}$	1	1	1						1	1	1	1	1	1						1
$I_{1,3}$	1	1				1	1	1												
$I_{3,1}$	1	1			1		1	1												

6

$I_{1,4}$	1	1			1	1		1	1					1	1	1				
$I_{4,1}$	1	1			1	1	1			1			1		1	1				
$I_{2,5}$			1	1			1			1	1	1	1	1			1	1		
$I_{5,2}$			1	1				1	1		1	1	1	1			1	1		
$I_{2,6}$			1	1					1	1		1	1	1					1	1
$I_{6,2}$			1	1					1	1	1		1	1					1	1
$I_{2,7}$			1	1			1	1	1	1	1		1							
$I_{7,2}$			1	1		1			1	1	1	1	1							
$I_{4,8}$		1				1	1									1				
$I_{8,4}$	1					1	1								1					
$I_{5,9}$	1								1	1								1		
$I_{9,5}$		1							1	1							1			
$I_{6,}$			1								1	1								1

I_1 0,6			1							1	1							1	
10																			

Number 1 in the matrix of Table 1 shows that links in a corresponding row and column cannot be active at the same time. Empty boxes in the matrix represent 0s.

5   A link bandwidth request is expressed in terms of the link capacity. Suppose that 64 credits are equivalent to full link capacity. If a link is given 64 credits, that link can be active all the time. If a link is given 32 credits, that link is active 50% of the time.

Suppose at a particular time, there exist the following bandwidth requests in credits:

10

$R_{0,2} = 35$

$R_{2,5} = 20$

$R_{2,6} = 15$

$R_{5,9} = 10$

15     $R_{6,10} = 10$

$R_{3,1} = 10$

$R_{1,0} = 10$

$R_{0,1} = 5$

20   The set of links requesting bandwidth is:

$L = \{\, l_{0,2}, l_{2,5}, l_{2,6}, l_{5,9}, l_{6,10}, l_{3,1}, l_{1,0}, l_{0,1} \,\}$

Using the interference matrix I, the degree of interference of $I_{0,2}$ in this set is computed as follows:

$$\alpha(I_{0,2}, L) = I[[I_{0,2}][I_{2,5}] + I[[I_{0,2}][I_{2,6}] + I[[I_{0,2}][I_{5,9}] + I[[I_{0,2}][I_{6,10}] + I[[I_{0,2}][I_{3,1}] + I[[I_{0,2}][I_{1,0}] + I[[I_{0,2}][I_{0,1}] = 5$$

Other degrees of interference can be computed similarly:

$$\alpha(I_{2,5}, L) = I[[I_{2,5}][I_{0,2}] + I[[I_{2,5}][I_{2,6}] + I[[I_{2,5}][I_{5,9}] + I[[I_{2,5}][I_{6,10}] + I[[I_{2,5}][I_{3,1}] + I[[I_{2,5}][I_{1,0}] + I[[I_{2,5}][I_{0,1}] = 3$$

$$\alpha(I_{2,6}, L) = I[[I_{2,6}][I_{0,2}] + I[[I_{2,6}][I_{2,5}] + I[[I_{2,6}][I_{5,9}] + I[[I_{2,6}][I_{6,10}] + I[[I_{2,6}][I_{3,1}] + I[[I_{2,6}][I_{1,0}] + I[[I_{2,6}][I_{0,1}] = 3$$

$$\alpha(I_{5,9}, L) = I[[I_{5,9}][I_{0,2}] + I[[I_{5,9}][I_{2,5}] + I[[I_{5,9}][I_{2,6}] + I[[I_{5,9}][I_{6,10}] + I[[I_{5,9}][I_{3,1}] + I[[I_{5,9}][I_{1,0}] + I[[I_{5,9}][I_{0,1}] = 2$$

$$\alpha(I_{6,10}, L) = I[[I_{6,10}][I_{0,2}] + I[[I_{6,10}][I_{2,5}] + I[[I_{6,10}][I_{2,6}] + I[[I_{6,10}][I_{5,9}] + I[[I_{6,10}][I_{3,1}] + I[[I_{6,10}][I_{1,0}] + I[[I_{6,10}][I_{0,1}] = 2$$

$$\alpha(I_{3,1}, L) = I[[I_{3,1}][I_{0,2}] + I[[I_{3,1}][I_{2,5}] + I[[I_{3,1}][I_{2,6}] + I[[I_{3,1}][I_{5,9}] + I[[I_{3,1}][I_{6,10}] + I[[I_{3,1}][I_{1,0}] + I[[I_{3,1}][I_{0,1}] = 1$$

$$\alpha(I_{1,0}, L) = I[[I_{1,0}][I_{0,2}] + I[[I_{1,0}][I_{2,5}] + I[[I_{1,0}][I_{2,6}] + I[[I_{1,0}][I_{5,9}] + I[[I_{1,0}][I_{6,10}] + I[[I_{1,0}][I_{3,1}] + I[[I_{1,0}][I_{0,1}] = 3$$

$$\alpha(I_{0,1}, L) = I[[I_{0,1}][I_{0,2}] + I[[I_{0,1}][I_{2,5}] + I[[I_{0,1}][I_{2,6}] + I[[I_{0,1}][I_{5,9}] + I[[I_{0,1}][I_{6,10}] + I[[I_{0,1}][I_{3,1}] + I[[I_{0,1}][I_{1,0}] = 4$$

5

10

15

20

25

9

Problem formulation

Assume that time division multiple access (TDMA) techniques are used to multiplex link activities. Given the constraints of the interference matrix and a list of bandwidth requests, attempt to find a schedule to make optimal use of total network capacity and fairly satisfy bandwidth requests.

An equivalent problem is to find an optimal schedule that satisfies all requests using the least amount of network resources, in this case, credits or time. If the average activity concurrency is defined as the average number of concurrent active links of a schedule, then the optimal schedule is the one having the highest average activity concurrency.

A schedule specifies when a set of links are active and also specifies the members of the set. Mathematically, a schedule S can be expressed as:

$S = \{(L_i, G_i) \mid G_i$ is the credits assigned to set of links L_i,

L_i is the set of links that can be all active at the same time without interfering with each other $\}$

<u>Example 2</u>

Continuing with Example 1 above, the following is one possible schedule for links requesting bandwidth:

$$S = \{ (\{ I_{5,9}, I_{6,10}, I_{3,1} \}, 10), (\{ I_{0,2} \}, 35), (\{ I_{2,6}, I_{0,1} \}, 5), (\{I_{2,6}, I_{1,0} \}, 10), (\{ I_{2,5} \}, 20) \}$$

This schedule uses 10+35+5+10+20 = 80 credits to satisfy 35+20+15+10+10+10+10+5 = 115 requested credits. The average activity concurrency is 115/80 = 1.4375

This schedule is not necessarily the best schedule for this example. In fact, using the algorithm described in detail below, one can find a better schedule using less credits while still satisfying all bandwidth requests.

An optimal schedule must satisfy the following conditions:

For any link, granted credits equals requested credits

$$\sum\nolimits_{Ijk \subset Li} G_i = R_{jk}$$

Minimal total network resource spent

$$(\sum G_i) <= (\sum G'_i) \text{ for } \forall S' = \{(L'_i, G'_i) \}$$

Because this problem is NP-hard, a heuristic algorithm is disclosed herein for a near optimal solution. For purposes of the discussion herein, a problem is NP-hard if an algorithm for solving it can be translated into one for solving any other NP-problem, *i.e.* nondeterministic polynomial time problem. NP-hard therefore means "at least as hard as any NP-problem," although it might, in fact, be harder.

Simulations show that in many cases this algorithm generates optimal schedules; and in cases that it does not, the schedules are usually close to optimal and are always better than average.

5   **Bandwidth allocation algorithm**

The algorithm disclosed herein is based on the assumption that there exists a centralized node (hub) in the network that coordinates all network activities (see Fig. 2). The hub 24 keeps the following data structures to represent its knowledge of the

10   network:

- Interference matrix 25 (defined above). It is important to note that interference matrix is symmetrical.

15   - Topology matrix 26: defines valid links that can transmit/receive data. This is a proper subset of the interference matrix.

- A list of credit request tokens 27. Each token represents a directional link that needs bandwidth.

20

Assume that each node 20, 22 in the network conveys its knowledge of interference, topology, and its bandwidth needs to the hub. The actual mechanism for transporting this information to the hub is within the knowledge of those skilled in the art and is, therefore, not discussed in detail herein. The hub collects this information from

individual nodes and constructs the interference matrix, topology matrix, and list of credit tokens to have a complete view of the network.

The bandwidth allocation algorithm running at hub is described as followed (see Fig. 3):

1. Sort credit request tokens in the descending order of the product of requested credits and degree of interference $\alpha(l_{ij}, L)$, where L is the set of links requesting for credits (100).

2. Pick the first token having a largest product (102). This is the first candidate of the set of links to be allocated credit for this round. Eliminate all other tokens from this round that cannot be active due to this link's activity (104).

3. Walk down the list and pick the next eligible token (106). This is the second candidate of the set of links to be allocated credits for this round. Eliminate all other tokens from this round that cannot be active due to this link's activity (108). Continue this step until the list is exhausted (110).

4. The result is a set of links that can be active at the same time $L_1 = \{ l_1, l_2, ...,l_n\}$ (112). Let β_{l_i} be requested credits of link l_i. The amount of credits allocated to each element of set L_1 is $\gamma_1 = \min\{\beta_{l1} , \beta_{l2} , ..., \beta_{ln} \}$. Adjust the requested credits for every element in L_1: $\beta_{l_i} = \beta_{l_i} - \gamma_1$ (114). Remove token(s) which have zero requested credits from the list of tokens (116).

5. Adjust the degree of interference of affected links, due to the fact that some tokens have been removed (118).

6. Repeat steps 1-5 until the list of tokens is empty (120).

7. The result is a list of (L_1, γ_1), (L_2, γ_2) .... (L_k, γ_k) (122). Now, prorate this list to attain the final schedule (124). Let S be the total resource of the network in terms of credit; and let $\chi_i = \gamma_i * S//\Sigma^{0,k} \gamma_j$. The list (L_1, χ_1), (L_2, χ_2) .... (L_k, χ_k) represents how the links are organized into sets of concurrent active links and how much resource each set of links are supposed to get. This list is broadcast to all nodes in the network (126).

Example 3

Use this algorithm to compute the schedule for Example 2.

Step 1 (see Table 2 below).

Table 2. Step 1

Link	Degree of interference $\alpha(l_{ij}, L)$	Requested credit R_{ij}	$\alpha(l_{ij}, L) * R_{ij}$
$l_{0,2}$	5	35	175
$l_{2,5}$	3	20	60
$l_{2,6}$	3	15	45
$l_{1,0}$	3	10	30
$l_{5,9}$	2	10	20
$l_{6,10}$	2	10	20

$I_{3,1}$	2	10	20
$I_{0,1}$	4	5	20

Steps 2-5:

Get the first Schedule S = { ({$I_{0,2}$, $I_{5,9}$, $I_{3,1}$}, 10) }

Go back to step 1 (see Table 3 below).

Table 3. Go Back to Step 1

Link	Degree of interference $\alpha(I_{ij}, L)$	Requested credit R_{ij}	$\alpha(I_{ij}, L) * R_{ij}$
$I_{0,2}$	5	25	125
$I_{2,5}$	2	20	40
$I_{2,6}$	2	15	30
$I_{1,0}$	2	10	20
$I_{6,10}$	2	10	20
$I_{0,1}$	2	5	10

Steps 2-5:

Get a revised Schedule S = { ({$I_{0,2}$, $I_{5,9}$, $I_{3,1}$}, 10), ({$I_{0,2}$}, 25) }

Go back to step 1 (see Table 4 below).

Table 4. Go Back t St p 1

Link	D gree of int rfer nc $\alpha(I_{ij}, L)$	Requ sted cr dit R_{ij}	$\alpha(I_{ij}, L) * R_{ij}$
$I_{2,5}$	1	20	20
$I_{2,6}$	1	15	15
$I_{1,0}$	1	10	10
$I_{6,10}$	1	10	10
$I_{0,1}$	1	5	5

Steps 2-5:

Get a revised Schedule:

5

$S = \{ (\{I_{0,2}, I_{5,9}, I_{3,1}\}, 10), (\{I_{0,2}\}, 25), (\{I_{2,5}, I_{1,0}, I_{6,0}\}, 10)\}.$

Go back to step 1 (see Table 5 below).

10

Table 5. Go Back to Step 1

Link	Degree of interference $\alpha(I_{ij}, L)$	Requested credit R_{ij}	$\alpha(I_{ij}, L) * R_{ij}$
$I_{2,6}$	1	15	15
$I_{2,5}$	1	10	10
$I_{0,1}$	0	5	0

Steps 2-5:

Get a revised Schedule:

15

16

$S = \{ (\{I_{0,2}, I_{5,9}, I_{3,1}\}, 10), (\{I_{0,2}\}, 25), (\{I_{2,5}, I_{1,0}, I_{6,0}\}, 10), (\{I_{2,6}, I_{0,1}\}, 5) \}$

Go back to step 1 (see Table 6 below)

Table 6. Go Back to Step 1

Link	Degree of interference $\alpha(I_{ij}, L)$	Requested credit R_{ij}	$\alpha(I_{ij}, L) * R_{ij}$
$I_{2,6}$	1	10	10
$I_{2,5}$	1	10	10

Steps 2-5:

Get a revised Schedule:

$S = \{ (\{I_{0,2}, I_{5,9}, I_{3,1}\}, 10), (\{I_{0,2}\}, 25), (\{I_{2,5}, I_{1,0}, I_{6,0}\}, 10), (\{I_{2,6}, I_{0,1}\}, 5), (\{I_{2,6}\}, 10) \}$.

Go back to step 1 (see Table 7 below).

Table 7. Go Back to Step 1

Link	Degree of interference $\alpha(I_{ij}, L)$	Requested credit R_{ij}	$\alpha(I_{ij}, L) * R_{ij}$
$I_{2,5}$	1	10	10

Steps 2-5:

Get the schedule:

$S = \{ (\{l_{0,2}, l_{5,9}, l_{3,1}\}, 10), (\{l_{0,2}\}, 25), (\{l_{2,5}, l_{1,0}, l_{6,0}\}, 10), (\{l_{2,6}, l_{0,1}\}, 5), (\{l_{2,6}\}, 10), (\{l_{2,5}\}, 10) \}.$

5   This schedule uses 10+25+10+5+10+10 = 70 credits to satisfy 35+20+15+10+10+10+10+5 = 115 requested credits. The average activity concurrency is 115/70 = 1.6428. Obviously, this schedule is better than the one presented in the previous example. In fact, it can be proved that this schedule is the optimal one for this particular example. There is no other schedule that can use less

10   number of credits to satisfy all these bandwidth requests.

Step 7:

Because the total resource is only 64 credits, the previous schedule is prorated to

15   obtain the final schedule:

$S_f = \{ (\{l_{0,2}, l_{5,9}, l_{3,1}\}, 9), (\{l_{0,2}\}, 23), (\{l_{2,5}, l_{1,0}, l_{6,0}\}, 9), (\{l_{2,6}, l_{0,1}\}, 5), (\{l_{2,6}\}, 9), (\{l_{2,5}\}, 9) \}.$

This schedule is broadcast to all nodes in the network.

20

Upon receiving the schedule, each node in the network uses the binary allocation map scheme to compute its own slot assignment. Allocation map is an array of numbers that is used to map a range of consecutive numbers to partially equally spaced numbers. The idea is that, given a portion of resources, a node can figure

25   out its active timeslots by projecting that portion (consecutive numbers) through the

map. For example, all links in set L_i are assigned to the range $[\Sigma^{0,i-1} \chi_j, \Sigma^{0,i} \chi_j]$, which, in turn, represent a set of near-equally spaced time slots.

Example 4

Assume that the allocation map is designed for 64 time slots, corresponding to 64 credits.

The allocation map for 64 time slots is shown in Table 8 below.

Table 8. Allocation Map, 64 Time Slots

tslot	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
index	1	33	17	49	9	41	25	57	5	37	21	53	13	45	29	61

tslot	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
index	3	35	19	51	11	43	27	59	7	39	23	55	15	47	31	63

tslot	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48
index	2	34	18	50	10	42	26	58	6	38	22	54	14	46	30	62

tslot	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64
index	4	36	20	52	12	44	28	60	8	40	24	56	16	48	32	64

A range of credit indices can be deduced for each set of links in the final schedule S_f. For example, the set $\{l_{0,2}, l_{5,9}, l_{3,1}\}$ is correspondent to [1,9]. Set $\{l_{0,2}\}$ is correspondent to [10,32]; and so on.

$S_f = \{ (\{l_{0,2}, l_{5,9}, l_{3,1}\}, 9), (\{l_{0,2}\}, 23), (\{l_{2,5}, l_{1,0}, l_{6,0}\}, 9), (\{l_{2,6}, l_{0,1}\}, 5), (\{l_{2,6}\}, 9), (\{l_{2,5}\}, 9) \}$

Using the combination of allocation map and the final schedule S_f, any node is aware of which link is active at a particular time slot t. For example, the set $\{l_{0,2}, l_{5,9}, l_{3,1}\}$ is active in time slots 1, 5, 9, 17, 25, 33, 41, 49, 57.

Maximizing network capacity using unscheduled time slots

To facilitate the explanation of using unscheduled time slots, use the schedule obtained in previous example.

$S = \{ (\{l_{0,2}, l_{5,9}, l_{3,1}\}, 10), (\{l_{0,2}\}, 25), (\{l_{2,5}, l_{1,0}, l_{6,0}\}, 10), (\{l_{2,6}, l_{0,1}\}, 5), (\{l_{2,6}\}, 10), (\{l_{2,5}\}, 10) \}$

Some notable points need to be made about this schedule:

1. The number of links in each set tends to be highest at the beginning of the schedule and tends to taper off toward the end of the schedule.

2. Even with the set causing most interference in the network, there are some links that can be active at the same time without causing interference to the links in the set.

3. The interference caused by sets at the beginning of the schedule tend to be the highest; and that interference tends to taper off going toward the end of the schedule.

5   With these observations, it can be seen the scheduled bandwidth very likely represents only about half of total network capacity. Hence, a collision-based mechanism is devised to use the other half, which is going to be wasted if not used otherwise.

10   Each node in the network maintains, for each of its local links, one set of links interfering with that link. Local links are links directly connected to the node. By using the schedule S broadcast by the Hub, a node knows which of its local links can be active without interfering with the scheduled links which are currently active. An active unscheduled link at time slot t is a link that is not scheduled to be active at

15   time t, but could be made active if the intended receiver is ready to receive. This can be decided by its directly connected nodes because this activity does not cause interference with the current active scheduled links. A link can be unscheduled at one time slot and is scheduled in another time slot. Active unscheduled links can interfere and collide which each other, but they do not interfere with the currently

20   active scheduled links.

Unscheduled links are mainly used when a node does not have uplink scheduled bandwidth and need to request bandwidth or need to send some small uplink transient traffic. It is used to boost up network capacity, as well as network response

25   time.

Example 5

Using Example 4, the final schedule is:

5    $S = \{ (\{l_{0,2}, l_{5,9}, l_{3,1}\}, 10), (\{l_{0,2}\}, 25), (\{l_{2,5}, l_{1,0}, l_{6,0}\}, 10), (\{l_{2,6}, l_{0,1}\}, 5), (\{l_{2,6}\}, 10), (\{l_{2,5}\}, 10) \}$

Pick one time slot t. Suppose that it corresponds to $(\{l_{2,6}, l_{0,1}\}, 5)$ in the schedule. This means that $l_{2,6}$ and $l_{0,1}$ are active at time slot t. The matrix of interference indicates

10   that any of links $\{ l_{4,8}\ l_{9,5} \}$ can also be active. Although each node does not maintain the matrix of interference for the whole network, it does keep sets of interference links for each of its local link. Hence, local nodes (4 and 9) know that they can activate the link at time slot t. In this specific example, if both $l_{4,8}\ l_{9,5}$ are active, they still do not collide. However, that is not always the case. Nodes can use a backoff

15   mechanism to resolve collision if it happens.

Although the invention is described herein with reference to the preferred embodiment, one skilled in the art will readily appreciate that other applications may be substituted for those set forth herein without departing from the spirit and scope of

20   the present invention. Accordingly, the invention should only be limited by the Claims included below.